REMARKS

Upon entry of the present amendment, claims 1-23 and 25-33 will be pending in the present application. Claims 1, 15, 22 and 32 are the independent claims. In the Official Action, dated January 4, 2006, claims 1-2, 4-13, 15, 17-30, 32 and 34 stand rejected under 35 U.S.C. § 103(a) as allegedly obvious over U.S. Patent No. 6,647,399 (Zaremba) in view of U.S. Patent No. 6,145,088 (Stevens). Claim 14 was rejected under 35 U.S.C. § 103(a) over Zaremba in view of Stevens, and further in view of U.S. Patent No. 6,038,379 (Fletcher et al.). Claims 3, 16, 31 and 33 were rejected under 35 U.S.C. § 103(a) over Zaremba in view of Stevens, and further in view of U.S. Patent No. 6,073,128 (Pongracz et al.).

Claims 22-34 have been amended herein to reflect the nature of Applicants' Figs. 4 and 5, as discussed below. Dependent claims 24 and 34 have been canceled without prejudice. No new matter has been added.

The outstanding rejections are respectfully traversed.

Summary of the Invention

Applicants respectfully request careful consideration of Figs. 4 and 5 and the accompanying description in the specification since various features of the invention seem misunderstood during the present examination. In this regard, it appears to the undersigned that Zaremba is being applied to each element of Applicants' claims independent of the recitations of the other elements and their interrelationships. Yet, a claim must be read as a whole, and accordingly, as discussed in detail below, nowhere is Zaremba believed to teach or suggest Applicants' invention in the full context of the recited claims (i.e., reading all of the limiations together).

To clarify, as described in connection with Applicants' Figure 4, the present invention describes the co-location of storage blocks of a target object to create an efficient cumulative backup object. In this case, a set of backup, incremental and cumulative files for a set of objects in the computing system are stored in memory blocks of a storage component 12 wherever located in the network.

Accordingly, since some changes during a time interval may be more sweeping than for other time intervals, more or less blocks B1, B2, B3, etc. will be implicated. Thus, in the example of Figure 4, a full backup file F1 is stored across eight memory blocks F1_B1 through F1_B8. The first incremental file F1_I1 at t2 is stored in one block F1_I1_B1, the second incremental file F1_I2 at t3 is stored across three blocks F1_I2_B1, F1_I2_B2 and F1_I2_B3, and so on. In accordance with the present invention, for a given target object **within the set of objects**, a cumulative file for a particular time may be created so that resource intensive searching through incremental files for the data necessary to restore a target object can be avoided at the time of restore, thereby significantly reducing the overall time needed to restore a target object from the time of its failure or corruption.

For example, it may be desirable to restore a target object, such as a word processing application and all of its dependent applications or objects, to a time t9. In accordance with the present invention, off-line calculation may be made wherein a file CT9 is created that represents the change of the target object (i.e., the target object designated within the set of objects) from a full backup F1 to time t9. Accordingly if the word processing application crashes or is corrupted, a restore operation may be made from CT9 in connection with F1. Because CT9 contains the storage block mapping information of the target object, ***only the***

storage blocks of the target object (of the set of objects) need be restored during the processing of F1 and CT9, thereby eliminating any restoring other than the target object.

Thus, in the example of Figure 4, if a target object had N storage blocks for restore purposes, and if the Nth storage block SN changed at time t3, the first storage block S1 changed at t6 and the second storage block changed at t8, a cumulative backup file for time t9 may be generated as follows. In an order from most recent to oldest (from t9 to t1), change of the target storage blocks S1, S2 ... SN is analyzed. When the last storage block for which change has occurred is discovered, the analysis is complete and the generation of CT9 may be achieved.

In the specific example shown in Fig. 4, first, the most recent change in S2 is discovered in F1_I7_B1 and copied into the second target object portion TOP_2. Then, the most recent change in S1 is discovered in F1_I5_B2 and copied into the first target object portion TOP_1. And so on, until the most recent change of the last storage block, SN, is discovered in F1_I2_B3 and copied into the last object portion TOP_N. No particular order of the storage blocks S1 to SN of the target object is necessary as long as all of the change of the target object is captured in the cumulative target object backup file. Thus, once CT9 has been generated, preferably off-line, and since CT9 includes the storage block mappings and the most recent change for the target object relative to the full backup file F1, a target object restore operation to t9 is simplified and streamlined having been tailored to the target object. Any incremental changes occurring subsequent to CT9, if any, would also be accommodated by a restore operation to a time subsequent to t9.

Zaremba

Zaremba relates to a backup operation of a target file, such as a database object. At least one operation is initiated to perform at least one delta backup of the target file to at least one delta backup set. A delta backup set includes changes made to a version of a full backup set of the target file generated at a first time. The at least one delta backup set is assigned a name component indicating the version of the full backup set generated at the first time subject to the delta backup and an operation is initiated to perform a full backup of the target file to a version of the full backup set at a second time. The second time follows the first time. A determination is made of all delta backup sets having the name component indicating the version of the full backup set generated at the first time during the full backup operation at the second time. An operation is initiated to render inactive the determined delta backup sets having the name component indicating the version of the full backup generated at the first time.

Independent Claims 1 and 15

Applicants concede that Zaremba discloses generating a full backup file for a set of objects ("A full backup set of a database object is provided the name of the database object, e.g., "DBNAME", with the extension "full," providing a name of "DBNAME.FULL." Col. 4, lines 14-17).

Applicants also concede that Zaremba discloses generating incremental files for the same database object for which a full backup set is created ("An incremental backup of a database object may have the name format DBOBJECTID.INCR.TIMESTAMP..." Col. 4, lines 21-27).

Applicants still further concede that Zaremba discloses generating a differential backup for the same database object ("A differential backup may have the naming convention – DBOBJECTID.DIFF, where DBOBJECTID is a unique identifier of the database object to backup..." Col. 4, lines 38-46).

However, such disclosure by Zaremba is believed to disclose no more than Applicants have already admitted as prior art in Applicants' Fig. 1D.

In this regard, Zaremba clearly includes no disclosure of identifying a target object within the set of objects for which at least full and incremental backups exist, and generating a cumulative backup file for that target object off-line, in the manner recited in claims 1 and similarly in claim 15.

Casting this point in the language of Zaremba itself, Zaremba includes no disclosure of *identifying a target object within the database object* for which full, incremental and differential backup files exist. Zaremba's database object and corresponding methodology described at Col. 4 is merely understood to disclose a naming convention for generating full, incremental and differential backup files for a database object, but there is no discussion of generating differential backup files for a target object within the database object based on at least the full and incremental backup files for the database object.

This is because Zaremba is really about a naming convention which enables the deletion of defunct full backup files (one only needs the most recent backup file), and the deletion of incremental and differential backup files that are based on the defunct full backup files. There simply is no discussion anywhere of generating cumulative files for a target object within Zaremba's database object in the manner recited by the present invention.

Reconsideration and withdrawal of the present rejection is thus respectfully requested.

Independent Claims 22 and 32

Applicants have amended claims 22 and 32, and associated dependent claims, to reflect similar language as claims 1 and 15 based on Applicants' Figs. 4 and 5, and associated discussion found in the specification. Claims 24 and 34 were canceled herein without prejudice.

In this respect, claims 22 and 32 are now believed patentable for reasons similar to claims 1 and 15, i.e., Zaremba is not believed to teach or suggest "a plurality of storage components for the storage of backup information for a plurality of target objects in the form of full, incremental and cumulative backup information, wherein the full and incremental backup information is associated with the collection of said plurality of target objects; and **wherein said cumulative backup information is generated off-line for a target object of the plurality of target objects based on the full and incremental backup information associated with the collection of said plurality of target objects ...,**" as now recited in claims 22 and 32.

CONCLUSION

Stevens does not cure the above-identified deficiency of root reference Zaremba with respect to Applicant's invention. Accordingly, neither Zaremba nor Stevens, taken alone or in combination, teach or suggest the claimed invention, as recited in claims 1, 15, 22 and 32. Claims 2-14, 16-21, 23, 25-31 and 33 depend from claims 1, 15, 22 and 32, respectively, and are believed allowable for the same reasons. Withdrawal of the rejection to claims 1-23 and 25-33 under 35 U.S.C. § 103(a) is respectfully requested.

Applicant believes that the present Amendment is responsive to each of the points raised by the Examiner in the Office Action, and submits that Claims 1-23 and 25-33 of the application are in condition for allowance. Favorable consideration and passage to issue of the application at the Examiner's earliest convenience is earnestly solicited.

Date: March 3, 2006

Thomas E. Watson
Registration No. 43,243

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439